Geometrische Datenstrukturen

Elmar Langetepe
University of Bonn

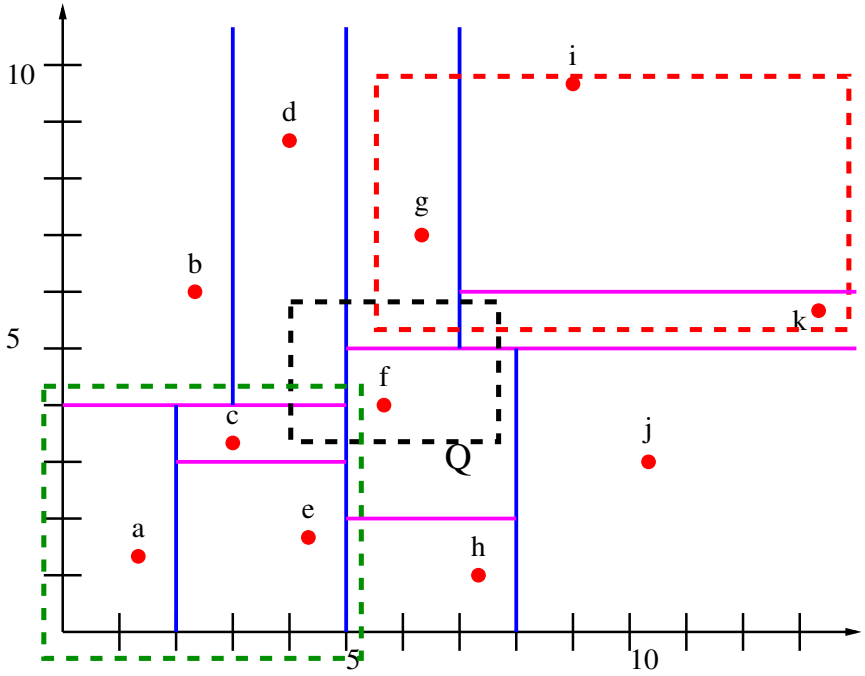
Query Laufzeit

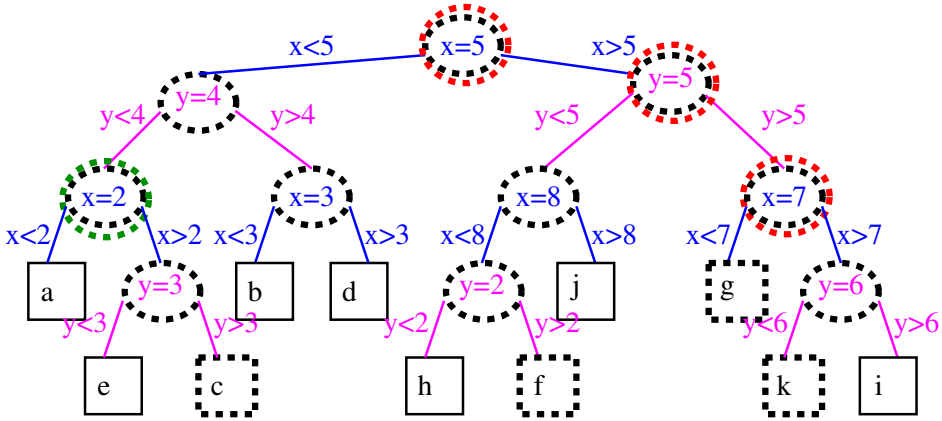
Query Laufzeit

Lemma 3.6: Sei T ein 2d-Baum der Höhe h mit n Punkten. Eine Bereichsanfrage mit achsenparallelen Rechteck Q läßt sich in Zeit $O\left(2^{\frac{h}{2}} + a\right)$ beantworten, wobei a die Größe der Antwort ist.

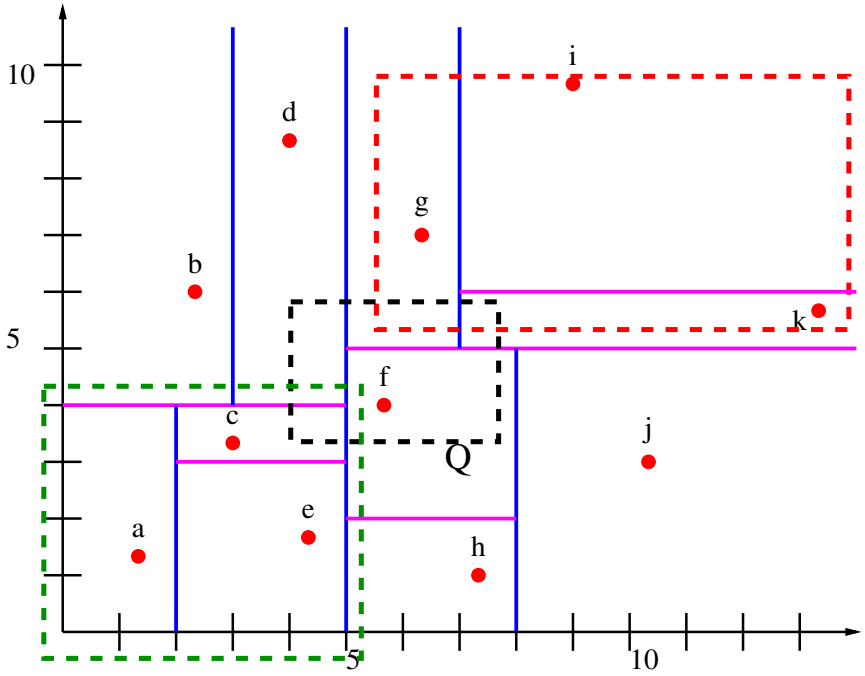
Beweis: Laufzeit $O\left(2^{\frac{h}{2}} + a\right)$

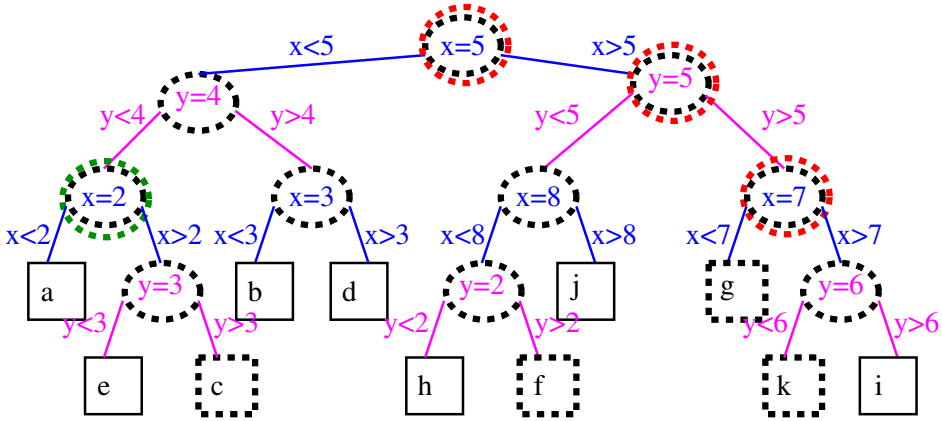
Beweis: Laufzeit $O\left(2^{\frac{h}{2}} + a\right)$





Beweis: Laufzeit $O\left(2^{\frac{h}{2}} + a\right)$





1. Knoten v mit $R(v) \subseteq q$
2. Knoten v mit $R(v) \not\subseteq q$ (aber $R(v) \cap q \neq \emptyset$), unterteilt sich in zwei Subtypen

Query Laufzeit

Lemma 3.6: Sei T ein 2d-Baum der Höhe h mit n Punkten. Eine Bereichsanfrage mit achsenparallelen Rechteck Q läßt sich in Zeit $O\left(2^{\frac{h}{2}} + a\right)$ beantworten, wobei a die Größe der Antwort ist.

Beweis:

- Zähle alle Knoten mit $R(v) \cap q \neq \emptyset$
- 1) Knoten v mit $R(v) \subseteq q$: Teilbaum ausgeben $a(v)$
- 2) Knoten v mit $R(v) \not\subseteq q$ (aber $R(v) \cap q \neq \emptyset$)
  - a) $q \subseteq R(v)$: Höhe h viele
  - b) Seite von q schneidet $R(v)$: $2^{\frac{k}{2}}$ viele für Tiefe k (Beweis)
- Aufsummieren über alle Tiefen k: $O\left(2^{\frac{h}{2}} + a\right)$

Ergebnis

Ergebnis

Theorem 3.7: Ein ausgeglichener 2d-Baum für n Punkte in der Ebene läßt sich in Zeit $O(n \log n)$ konstruieren. Er benötigt $O(n)$ Speicherplatz . Eine Bereichsanfrage mit achsenparallelen Rechteck q kann in Zeit $O\left(\sqrt{n} + a\right)$ beantwortet werden, wobei a die Größe der Antwort ist.

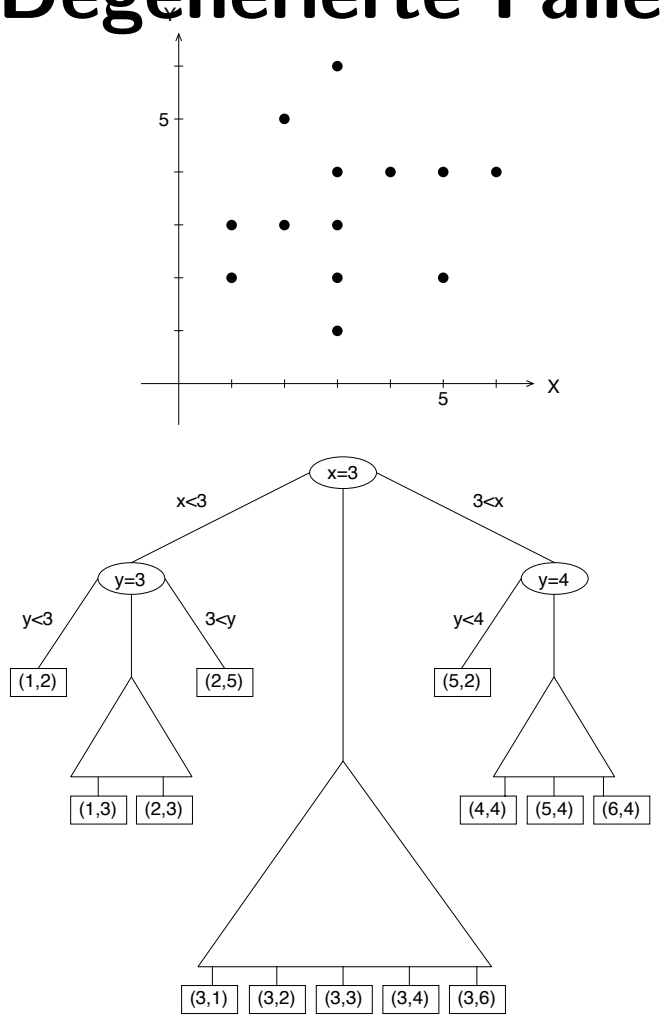
Ergebnis

Theorem 3.7: Ein ausgeglichener 2d-Baum für n Punkte in der Ebene läßt sich in Zeit $O(n \log n)$ konstruieren. Er benötigt $O(n)$ Speicherplatz . Eine Bereichsanfrage mit achsenparallelen Rechteck q kann in Zeit $O\left(\sqrt{n} + a\right)$ beantwortet werden, wobei a die Größe der Antwort ist.

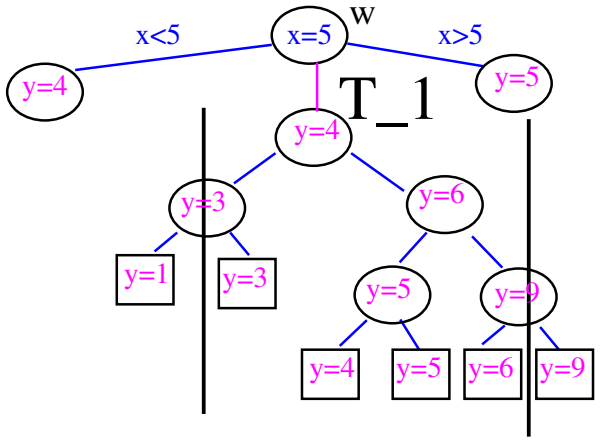
Sortieren nach X und Y und rekursiv in gleichgroße Teilmengen aufteilen, lineares Aufteilen

Degenerierte Fälle

Degenerierte Fälle



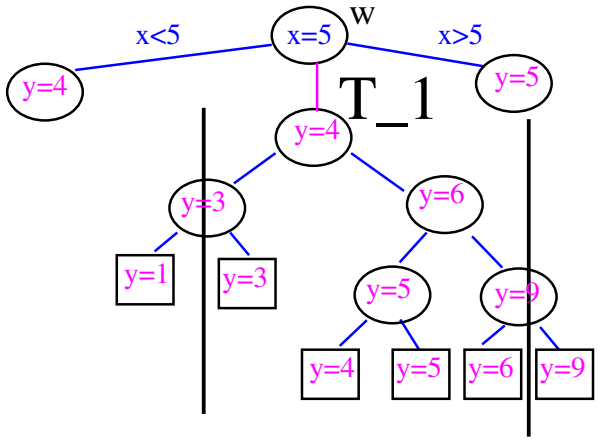
Laufzeitabschätzung



q mit Y=7 und Y=2

Laufzeitabschätzung

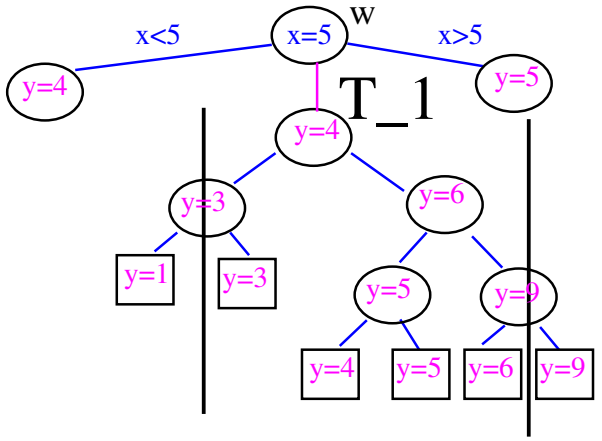
- Zusätzlicher Aufwand für dritte Struktur



q mit Y=7 und Y=2

Laufzeitabschätzung

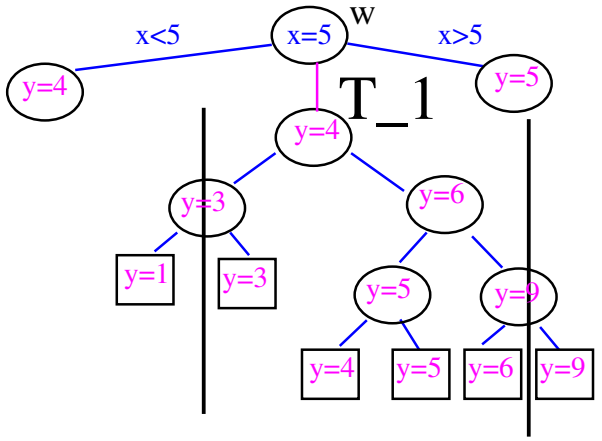
- Zusätzlicher Aufwand für dritte Struktur
- $R(v)$ in T_1 ist Intervall: $I(v)$



q mit Y=7 und Y=2

Laufzeitabschätzung

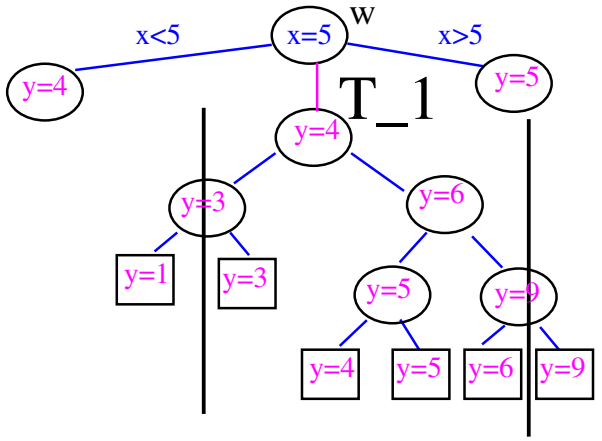
- Zusätzlicher Aufwand für dritte Struktur
- $R(v)$ in T_1 ist Intervall: $I(v)$
- Blätter von T_1 für Knoten Tiefe k: $\leq \frac{n}{2^{k-1}}$



q mit Y=7 und Y=2

Laufzeitabschätzung

- Zusätzlicher Aufwand für dritte Struktur
- $R(v)$ in T_1 ist Intervall: $I(v)$
- Blätter von T_1 für Knoten Tiefe k: $\leq \frac{n}{2^{k-1}}$
- Beweis



q mit Y=7 und Y=2

Ergebnis entartete 2d-Bäume

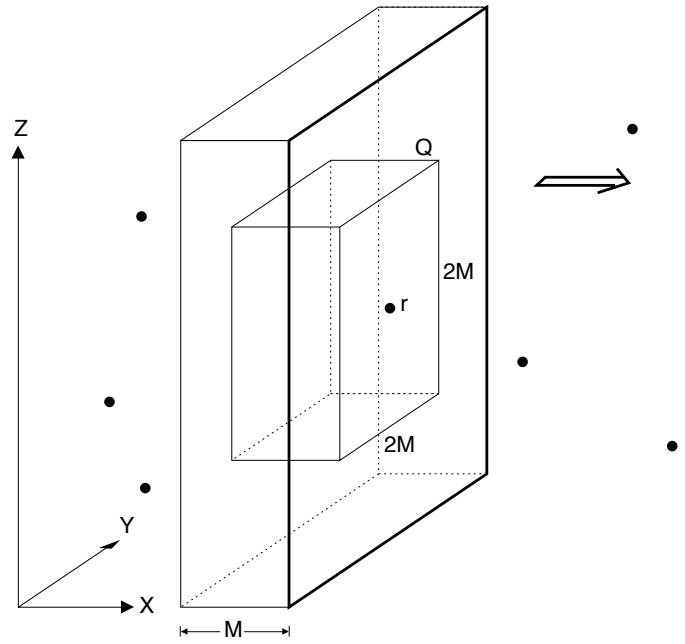
Ergebnis entartete 2d-Bäume

Theorem 3.7 (entartet): Ein ausgeglichener 2d-Baum für n Punkte in der Ebene läßt sich in Zeit $O(n \log n)$ konstruieren. Er benötigt $O(n)$ Speicherplatz. Eine Bereichsanfrage mit achsenparallelen Rechteck q kann in Zeit $O\left(\sqrt{n} + a\right)$ beantwortet werden, wobei a die Größe der Antwort ist.

Anwendung 3D Sweep

Anwendung 3D Sweep

- Sweep im Raum, Closest-Pair: Laufzeit $O(n(e(n) + b(n)))$
- 2d-Baum: $b(n) \in O(\sqrt{n} + a)$, $e(n) \in O(\log n)$



Verallgemeinerung

Verallgemeinerung

Theorem 3.7 (entartet): Ein ausgeglichener kd-Baum für n Punkte im $\mathbb{R}^k$ läßt sich in Zeit $O(n \log n)$ konstruieren. Er benötigt $O(n)$ Speicherplatz. Eine orthogonale Bereichsanfrage kann in Zeit $O\left(n^{1-\frac{1}{k}} + a\right)$ beantwortet werden, wobei a die Größe der Antwort ist.

Verallgemeinerung

Theorem 3.7 (entartet): Ein ausgeglichener kd-Baum für n Punkte im $\mathbb{R}^k$ läßt sich in Zeit $O(n \log n)$ konstruieren. Er benötigt $O(n)$ Speicherplatz. Eine orthogonale Bereichsanfrage kann in Zeit $O\left(n^{1-\frac{1}{k}} + a\right)$ beantwortet werden, wobei a die Größe der Antwort ist.

k zyklische Splitebenen, ternäre Struktur, Datenobjekte in Split(hyper)ebene in $(k-1)d$-Baum ablegen

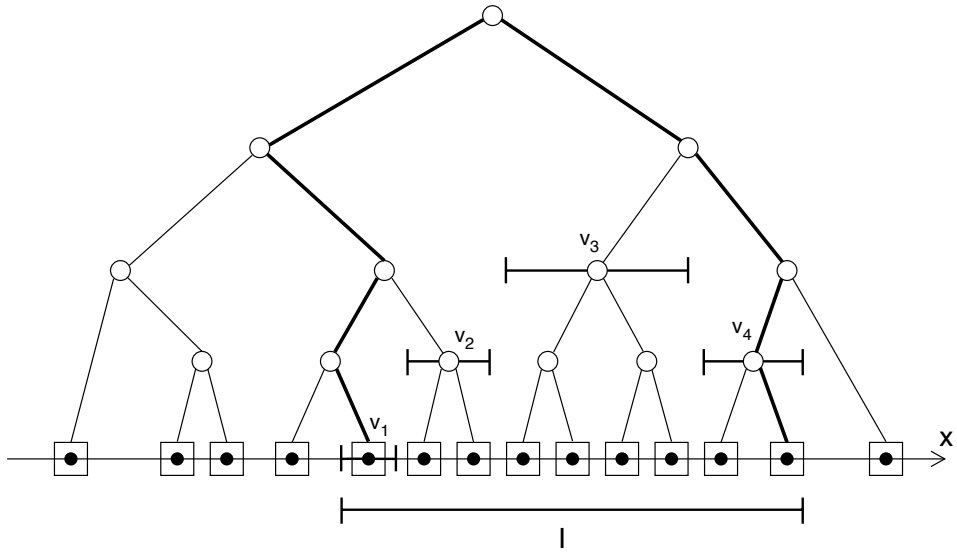
Verallgemeinerung

Theorem 3.7 (entartet): Ein ausgeglichener kd-Baum für n Punkte im $\mathbb{R}^k$ läßt sich in Zeit $O(n \log n)$ konstruieren. Er benötigt $O(n)$ Speicherplatz. Eine orthogonale Bereichsanfrage kann in Zeit $O\left(n^{1-\frac{1}{k}} + a\right)$ beantwortet werden, wobei a die Größe der Antwort ist.

k zyklische Splitebenen, ternäre Struktur, Datenobjekte in Split(hyper)ebene in $(k-1)d$-Baum ablegen

kd-Baum: Platzoptimal, Laufzeitverbesserungen Query: $O(a + \log^2 n)$

Range Tree (Bereichsbaum)



Eindimensionaler Bereichsbaum: Intervalle die ein Anfrageintervall ausschöpfen

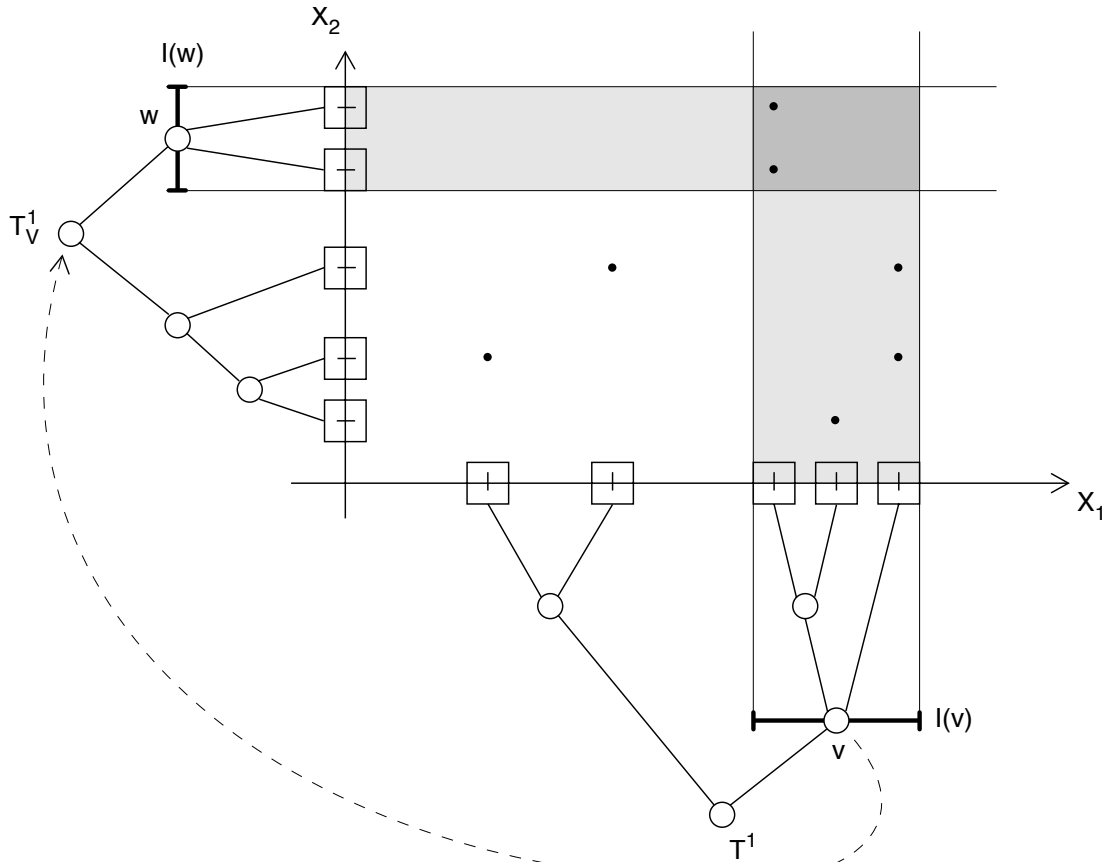
Jeder Knoten beschreibt ein Intervall

Knoten min. Höhe, Intervall disjunkt überdecken: $O(\log n)$ viele

Rekursiv k-dimensionaler Range Tree T^k

- Rekursiv Bereichsbaum T^k mit Dim. k
- T^1 für $D^1 = \{x_1 | (x_1, x_2, \ldots, x_k) \in D\}$
- Für jeden Knoten v in T^1: Konstr. $(k-1)$-dim Bereichsbaum T_v^{k-1} der Menge $D_v^{k-1} = \{(x_2, x_3, \ldots, x_k) | x_1 \in I(v)\}$
- Zeiger bei v zeigt auf T_v^{k-1}
- Letzter Baum enthält im Blatt alle Knoten aus D

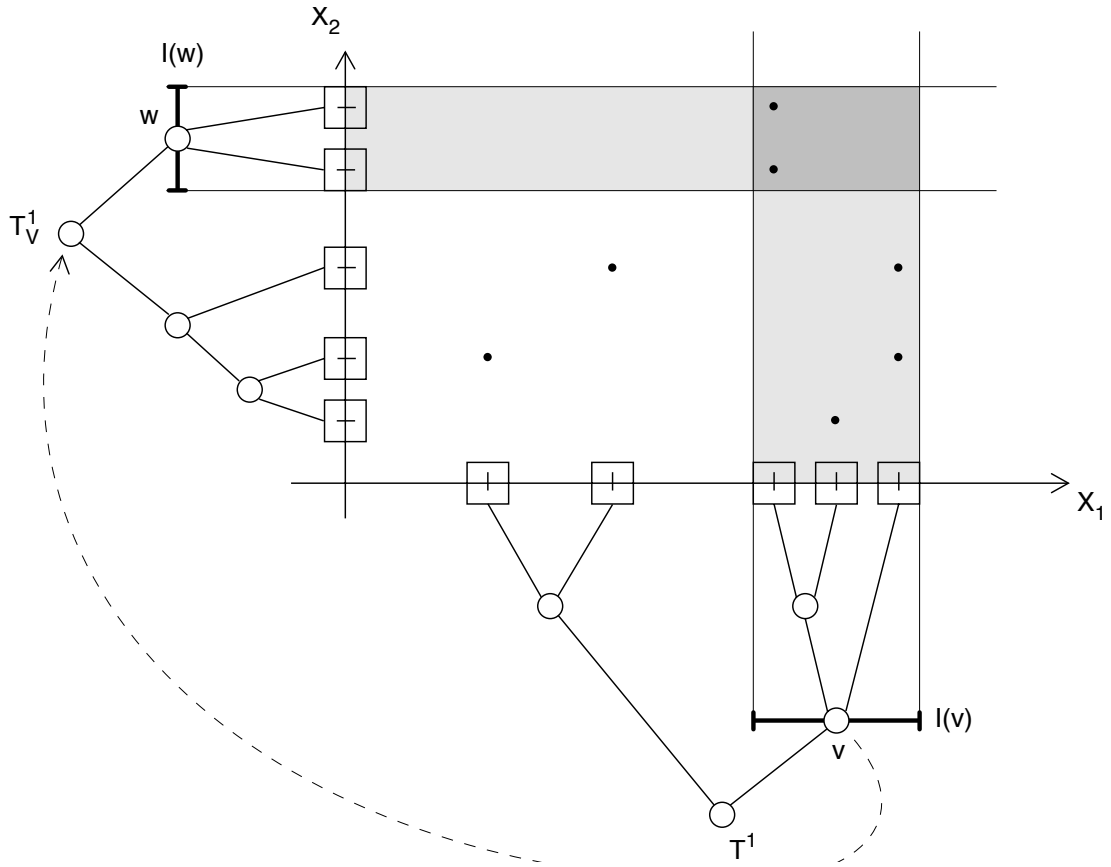
Beispiel 2-dimensionaler Range Tree



Query: k-dimensionaler Range Tree

- Hyperrechteck $q \in \mathbb{R}^k$: $q = (I_1, I_2, \ldots, I_k)$
- $k = 1$: Beantworte die Frage im eindim. Suchbaum
- Sonst: Bestimme Knoten $v_1, \ldots, v_l$ in T^1:
  Intervalle $I(v_i)$ schöpfen zusammen I_1 aus
  $v_1, \ldots, v_l$ haben minimale Höhe mit dieser Eigenschaft
- Für $1 \leq i \leq l$ beantworte die Anfrage $q = (I_2, \ldots, I_k)$ für $T^{k-1}_{v_i}$
- Korrekt: Zuerst unter X_1-Koordinaten suchen
  Von diesen unter X_2-Koordinaten suchen
  Uswusf.

Beispiel 2-dimensionaler Range Tree



k-dimensionaler Range Tree

Theorem 3.11: Ein k-dimensionaler Bereichsbaum für n Punkte im $\mathbb{R}^k$ kann in Zeit $O(n(\log n)^{k-1})$ mit Platz $O(n(\log n)^{k-1})$ aufgebaut werden. Eine Bereichsanfrage mit Hyperrechteck $q \subset \mathbb{R}^k$ kann in Zeit $O(a + (\log n)^k)$ beantwortet werden. Dabei ist a die Größe der Antwort.

Beweis: Speicherplatz Punkt $p = (x_1, x_2, \ldots, x_k)$

- Baum T^1:
  1) Im Baum T^1 einmal
  2) In max. $\log n$ vielen Intervallen $I(v_i)$ von T^1
- Bäume $T^{k-1}_{v_i}$, $i = 1, \ldots, l$
  1) In Baum $T^{k-1}_{v_i}$ einmal
  2) In max. $\log n$ vielen Intervallen $I(w_j)$ von $T^{k-1}_{v_i}$

- Insgesamt: $\sum_{i=0}^{k-1}(\log n)^i \in O((\log n)^{k-1})$
- Alle Punkte $O(n(\log n)^{k-1})$

k-dimensionaler Range Tree

Theorem 3.11: Ein k-dimensionaler Bereichsbaum für n Punkte im $\mathbb{R}^k$ kann in Zeit $O(n(\log n)^{k-1})$ mit Platz $O(n(\log n)^{k-1})$ aufgebaut werden. Eine Bereichsanfrage mit Hyperrechteck $q \subset \mathbb{R}^k$ kann in Zeit $O(a + (\log n)^k)$ beantwortet werden. Dabei ist a die Größe der Antwort.

Beweis: Query! $q = (I_1, I_2, \ldots, I_k)$

- T^1: Intervalle $I(v_i)$ die I_1 ausschöpfen max. $2 \log n$ viele
- Induktiv: $O((\log n_i)^{k-1} + a_i)$ für $T^{k-1}_{v_i}$, Ind. Anfang: $k = 2$ klar, Ind. Schluss:

  Bereichsanfrage für alle $T^{k-1}_{v_i}$:

  $C \cdot \log n + \sum_{i=1}^{2 \log n} (\log n_i)^{k-1} + a_i \in O((\log n)^k + a)$

  Punkte in den Intervallen $I(v_i)$ disjunkt: $\sum a_i = a$